

Received September 6, 2019, accepted September 16, 2019, date of publication September 18, 2019,
date of current version October 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2942169

Solving Large-Scale Function Optimization Problem by Using a New Metaheuristic Algorithm Based on Quantum Dolphin Swarm Algorithm

WEIBIAO QIAO^{1,2} AND ZHE YANG³

¹School of Environmental and Municipal Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, China

²Sinopec Petroleum Engineering Zhongyuan Company Ltd., Zhengzhou 450046, China

³School of Computer Science, The University of Manchester, Manchester M13 9PL, U.K.

Corresponding author: Weibiao Qiao (wbq0408@163.com)

This work was supported by the High-Level Talents Start-Up Project of the North China University of Water Resources and Electric Power under Grant 40691.

ABSTRACT Meta-heuristic algorithm has been a research hotspot in solving the optimal solution of large-scale functions. However, meta-heuristic algorithms are prone to fall into local optimum problems, such as the recently proposed dolphin swarm algorithm (DSA). To solve this problem, in this study, the quantum search algorithm is introduced into DSA. In addition, to test the performance of the proposed quantum dolphin swarm algorithm (QDSA), six commonly used large-scale functions (e.g. Rotated hyper-ellipsoid function) are taken as examples. Furthermore, some advanced algorithms (e.g. whale optimization algorithm (WOA)) are used for comparison. The results show that the ability of QDSA to obtain global optimal solution is obviously improved compared with DSA, and the performance of QDSA is superior to other algorithms considered for comparison. Finally, it can be concluded that such a novel meta-heuristic algorithm may help to improve the problem of solving the optimal solution of large-scale functions.

INDEX TERMS Quantum search algorithm, dolphin swarm algorithm, large-scale function optimization.

I. INTRODUCTION

The optimization problem of large-scale functions has always been a hotspot in the field of optimization. Large-scale function optimization problems have important applications in theory and engineering [1], [2]. Many practical engineering problems can be solved by transforming them into function optimization problems, such as multi-parameter function optimization [3]–[5]. In this study, we define a function whose variable is higher than 50 as a large-scale function. Therefore, it is of great significance to study the solution of large-scale functions.

However, the large-scale function is unable to draw intuitive function image in three-dimensional space, and with the increase of function dimension, the scale of search space presents exponential growth, which brings some difficulties to the optimization of large-scale function [6]–[8].

The associate editor coordinating the review of this manuscript and approving it for publication was Dimitrios Katsaros^{1,2}.

Therefore, some scholars have tried to use the meta-heuristic algorithm to solve the optimization problem of large-scale functions in recent years, these research results are shown in TABLE 1.

From TABLE 1, it is clear that some meta-heuristic algorithms proposed in the past few years are inspired by particle swarm optimization (PSO). PSO has been proposed for several years, but it is still easy to fall into local optimum. What is more, these meta-heuristic algorithms in TABLE 1 only improve this problem to a certain extent. Therefore, some improved meta-heuristic algorithms are established, as shown in TABLE 2.

For TABLE 2, some improved meta-heuristic algorithms are proposed, but these improved meta-heuristic algorithms just can solve the low-dimensional function, for example, some functions have two or three variables. So, in order to solve the optimization problem of large-scale functions, it is necessary to establish a new meta-heuristic algorithm.

TABLE 1. Some meta-heuristic algorithms for solving high-dimensional function optimization problem in recent years.

Algorithm name	Occurrence	Results and conclusions	Reference
Artificial immune system	2005	The results indicate that immune memory clonal programming algorithm is shown to be capable of solving high-dimensional function optimization, comparing with breeder genetic algorithm.	[9]
Differential evolution algorithm	2007	The results show that DE variants etc. have excellent performance based on the multiple test functions.	[10]
A fast bacterial swarming algorithm	2008	The proposed fast bacterial swarming algorithm has the improved optimization accuracy.	[11]
A small world algorithm	2009	The experimental results show that decimal-coding small world optimization algorithm can accomplish the satisfied solution	[12]
A new cooperative coevolution orthogonal artificial bee colony algorithm	2013	Cooperative coevolution orthogonal artificial bee colony solving high-dimensional function optimization problems, which is effective.	[13]
Differential evolution algorithm	2014	The results show that the new differential evolution is effective for large-scale optimization.	[14]
Grasp algorithm	2015	Experimental results show the supremacy of the proposed method is superior.	[15]
Particle swarm optimization algorithm	2015	Particle swarm algorithm generates better results in multi-peak problems	[16]
Partial differential equations and deep learning	2017	Numerical results suggest that the established algorithm is effective.	[17]
Surrogate-assisted hierarchical particle swarm optimization	2018	Their experimental results demonstrate that the proposed method is competitive	[18]
Inspired grey wolf optimization algorithm	2018	The results show that the proposed algorithm finds more accurate solutions.	[19]
Incremental gravitational search algorithm	2018	IGSA is better than GSA.	[20]

TABLE 2. Some improved meta-heuristic algorithms in recent years.

Algorithm name	Occurrence	Results and conclusions	Reference
Particle swarm ant colony optimization	2007	The results demonstrate the effectiveness of the proposed particle swarm ant colony optimization method.	[21]
Improving particle swarm optimization performance with local search	2010	Simulation results demonstrate that the Improving particle swarm optimization performance with local search is more stable than several other advanced methods.	[22]
Particle swarm optimization gravitational search algorithm	2014	The experimental results show a better performance of particle swarm optimization gravitational search algorithm.	[23]
Particle swarm and artificial bee colony	2015	Results demonstrate that the particle swarm and artificial bee colony algorithm is a robust optimization method for solving high-dimensional optimization problems.	[24]
Modified differential evolution whale optimization algorithm	2018	The results show that the modified differential evolution whale optimization algorithm has better performance.	[25]
Particle swarm grey wolf optimization algorithm	2018	The simulation results show that the proposed algorithm can better search the global optimal solutions and better robustness than some excellent algorithms.	[26]
Hybrid firefly particle swarm optimization	2018	The simulation results show that the hybrid firefly particle swarm optimization algorithm can better robustness than other algorithm.	[27]
Genetic particle swarm optimization algorithm	2018	The solution results show that the proposed genetic particle swarm optimization algorithm provides fast optimization solutions.	[28]
Simulated annealing moth flame optimization	2018	The results show competitive results of simulated annealing moth flame optimization concerning other meta-heuristic algorithms.	[29]
Wolf pack search local search	2019	The test of benchmarks shows that the results are closer to the theoretical optimal values.	[30]
Chaotic dolphin swarm algorithm	2019	It is concluded that such a new meta-heuristic algorithm could help to improve the shortcomings of DSA and increase the applied range of DSA.	[31]

In 2016, dolphin swarm algorithm (DSA) is proposed and applied by [32]. However Similar to other metaheuristic algorithms, DSA has the problem of an optimal balance between exploitation and exploration and of easy falling into local optimum. So, to address this problem, a new algorithm named quantum dolphin swarm algorithm (QDSA) is proposed by introducing quantum search algorithm into dolphin swarm algorithm in this paper.

II. DOLPHIN SWARM ALGORITHM (DSA)

A. HUNTING PROCESS OF DOLPHIN SWARM

The development of some meta-heuristic algorithms is inspired by PSO. DSA is no exception. Wu et al. observed

dolphin swarm and found some behaviors, including echolocation, division of labor, cooperation and information exchange. Dolphins catch and feed on their prey through these behavior patterns.

B. SOME BASIC DEFINITIONS

1) DOLPHIN

In the optimization process of DSA, each dolphin is equivalent to the particle in the PSO, representing a feasible solution in the optimization problem. In this paper, dolphins are defined as $Dol_i = [x_1, x_2, \dots, x_D]^T (i = 1, 2, \dots, N)$, where N means the number of dolphins, and $x_j (j = 1, 2, \dots, D)$ mean the component.

2) OPTIMAL INDIVIDUAL AND NEIGHBORHOOD SOLUTION

The two important definitions related to the DSA are the optimal individual solution (expressed as \mathbf{L}) and neighborhood solution (shown as \mathbf{K}). Moreover, for each $\mathbf{Dol}_i (i = 1, 2, \dots, N)$, there are two important parameters which are $\mathbf{K}_i (i = 1, 2, \dots, N)$ and $\mathbf{L}_i (i = 1, 2, \dots, N)$, where \mathbf{L}_i means the optimal solution that \mathbf{Dol}_i finds in a unique time, and \mathbf{K}_i represents the optimal solution of what \mathbf{Dol}_i gets from others.

3) FITNESS AND DISTANCE

In DSA, three kinds of distances need to be defined, which are the distance between \mathbf{L}_i and \mathbf{K}_i , named DLK_i , the distance between \mathbf{Dol}_i and \mathbf{K}_i , named DK_i , and the distance between \mathbf{Dol}_i and \mathbf{Dol}_j , named $DD_{i,j}$. These three distances are expressed as follows:

$$DD_{i,j} = \|\mathbf{Dol}_i - \mathbf{Dol}_j\|, \quad i, j = 1, 2, \dots, N, i \neq j. \quad (1)$$

$$DK_i = \|\mathbf{Dol}_i - \mathbf{K}_i\|, \quad i = 1, 2, \dots, N. \quad (2)$$

$$DKL_i = \|\mathbf{L}_i - \mathbf{K}_i\|, \quad i = 1, 2, \dots, N. \quad (3)$$

C. CRITICAL STAGES

1) SEARCH STAGE

When dolphins are searching for prey, they deliver the sound in M directions in the area near the dolphin in general. To accurately describe the process of dolphin hunting for prey, the sound is defined as $\mathbf{V}_i = [v_1, v_2, \dots, v_D] (i = 1, 2, \dots, M)$ in this study, where $v_j (j = 1, 2, \dots, D)$ means the component of each dimension, called M means the number of sounds and the direction attribute of the sound. Furthermore, sound satisfies $\|\mathbf{V}_i\| = \text{speed} (i = 1, 2, \dots, M)$, where ‘speed’ is the speed attribute of sound. A maximum search time T_1 is set to prevent dolphins from falling into the search phase. In the range of 0 to T_1 , the sound \mathbf{V}_j that $\mathbf{Dol}_i (i = 1, 2, \dots, N)$ makes at time t will find a new solution X_{ijt} . The definition of X_{ijt} is as follows.

$$X_{ijt} = \mathbf{Dol}_i + \mathbf{V}_j \quad (4)$$

For X_{ijt} , its fitness value E_{ijt} is defined as follows:

$$E_{ijt} = \text{Fitness}(X_{ijt}). \quad (5)$$

If

$$\begin{aligned} E_{iab} &= \min_{j=1,2,\dots,M; t=1,2,\dots,T_1} E_{ijt} \\ &= \min_{j=1,2,\dots,M; t=1,2,\dots,T_1} \text{Fitness}(X_{ijt}) \end{aligned} \quad (6)$$

Then \mathbf{L}_i of \mathbf{Dol}_i is defined as

$$\mathbf{L}_i = \mathbf{X}_{iab} \quad (7)$$

If

$$\text{Fitness}(\mathbf{L}_i) < \text{Fitness}(\mathbf{K}_i) \quad (8)$$

Then \mathbf{L}_i replaces \mathbf{K}_i ; otherwise, \mathbf{K}_i does not vary.

After all the $\mathbf{Dol}_i (i = 1, 2, \dots, N)$ update, \mathbf{L}_i and \mathbf{K}_i , dolphins get into the call stage.

2) RECEPTION STAGE

In DSA, the reception stage happens after the call stage. The reception stage is first represented in detail.

An $N \times N$ -order matrix which is named ‘transmission time matrix’ ($\mathbf{TS} = (\mathbf{TS}_{ij} (i = 1, 2, \dots, N; j = 1, 2, \dots, N))$) is used to express the information exchange between dolphins, where \mathbf{TS}_{ij} is the rest of the time for the sound of moving from \mathbf{Dol}_i to \mathbf{Dol}_j . When dolphins enter the reception stage, that all components $\mathbf{TS}_{ij} (i = 1, 2, \dots, N; j = 1, 2, \dots, N)$ in the \mathbf{TS} will decrease indicate that the sounds spread on any component \mathbf{TS}_{ij} in the \mathbf{TS} , and if

$$\mathbf{TS}_{i,j} = 0 \quad (9)$$

This shows that the sound, which will be obtained by \mathbf{Dol}_i , transmitting from \mathbf{Dol}_j to \mathbf{Dol}_i . What is more, \mathbf{TS}_{ij} will be substituted by a new acquisition time, which is named ‘maximum transmission time’ (T_2). Through this process, the related sound will be received. Besides, comparing \mathbf{K}_i and \mathbf{K}_j , if

$$\text{Fitness}(\mathbf{K}_i) > \text{Fitness}(\mathbf{K}_j) \quad (10)$$

Then \mathbf{K}_i will be replaced by \mathbf{K}_j , or \mathbf{K}_i does not vary. Then, DSA enters the predation stage.

3) CALL STAGE

For \mathbf{K}_i , \mathbf{K}_j , and $\mathbf{TS}_{i,j}$, if

$$\text{Fitness}(\mathbf{K}_i) > \text{Fitness}(\mathbf{K}_j) \quad (11)$$

$$\mathbf{TS}_{i,j} > \left\lceil \frac{DD_{i,j}}{A \cdot \text{speed}} \right\rceil \quad (12)$$

where A means the acceleration. Then, $\mathbf{TS}_{i,j}$ is updated in the light of the following equation:

$$\mathbf{TS}_{i,j} = \left\lceil \frac{DD_{i,j}}{A \cdot \text{speed}} \right\rceil \quad (13)$$

After all the $\mathbf{TS}_{i,j}$ is updated, DSA enters the reception stage.

4) PREDATION STAGE

In this stage, each dolphin hunts for preys within a surrounding radius called R_2 . Besides, R_2 determines the distance between its position after the predation obtains a new position and the dolphin’s optimal neighborhood solution. In addition, the search radius R_1 , which is the maximum range in the search stage, is calculated as follows:

$$R_1 = T_1 \times \text{speed} \quad (14)$$

Then, $\mathbf{Dol}_i (i = 1, 2, \dots, N)$ is supposed to an example for depicting the calculation of R_2 and update the dolphin’s position.

(a) For $\mathbf{Dol}_i (i = 1, 2, \dots, N)$, if

$$DK_i \leq R_1 \quad (15)$$

Next, R_2 is calculated on the basis of EQUATION (16).

$$R_2 = \left(1 - \frac{2}{e}\right) DK_i, \quad e > 2 \quad (16)$$

where e means the radius reduction coefficient.

After obtaining R_2 , \mathbf{Dol}_i 's new position \mathbf{newDol}_i is got:

$$\mathbf{newDol}_i = \mathbf{K}_i + \frac{\mathbf{Dol}_i - \mathbf{K}_i}{\mathbf{DK}_i} R_2. \quad (17)$$

(b) For \mathbf{Dol}_i ($i = 1, 2, \dots, N$), if

$$\mathbf{DK}_i > R_1 \quad (18)$$

and

$$\mathbf{DK}_i \geq \mathbf{DKL}_i \quad (19)$$

Next, R_2 is calculated on the basis of EQUATION (20).

$$R_2 = \left(1 - \frac{\frac{\mathbf{DK}_i}{\text{Fitness}(\mathbf{K}_i)} + \frac{\mathbf{DKL}_i - \mathbf{DK}_i}{\text{Fitness}(\mathbf{L}_i)}}{e \cdot \mathbf{DK}_i \frac{1}{\text{Fitness}(\mathbf{K}_i)}} \right) \mathbf{DK}_i, e > 2 \quad (20)$$

After obtaining R_2 , \mathbf{Dol}_i 's new position \mathbf{newDol}_i can be obtained:

$$\mathbf{newDol}_i = \mathbf{K}_i + \frac{\mathbf{Random}}{\|\mathbf{Random}\|} R_2 \quad (21)$$

(c) For \mathbf{Dol}_i ($i = 1, 2, \dots, N$), if it satisfies EQUATION (18) and

$$\mathbf{DK}_i < \mathbf{DKL}_i \quad (22)$$

Next, R_2 is calculated on the basis of EQUATION (23).

$$R_2 = \left(1 - \frac{\frac{\mathbf{DK}_i}{\text{Fitness}(\mathbf{K}_i)} + \frac{\mathbf{DKL}_i - \mathbf{DK}_i}{\text{Fitness}(\mathbf{L}_i)}}{e \cdot \mathbf{DK}_i \frac{1}{\text{Fitness}(\mathbf{K}_i)}} \right) \mathbf{DK}_i, e > 2 \quad (23)$$

After obtaining R_2 , \mathbf{Dol}_i 's new position \mathbf{newDol}_i is got by EQUATION (21).

After \mathbf{Dol}_i moves to the position \mathbf{newDol}_i , comparing \mathbf{newDol}_i with \mathbf{K}_i in terms of fitness, if

$$\text{Fitness}(\mathbf{newDol}_i) > \text{Fitness}(\mathbf{K}_i) \quad (24)$$

Then \mathbf{K}_i will be replaced by \mathbf{newDol}_i , or \mathbf{K}_i does not vary.

At last, If the end condition of iteration is satisfied, DSA gets into the termination stage, otherwise, DSA gets into the search stage.

III. QUANTUM DOLPHIN SWARM ALGORITHM (QDSA)

In quantum space [33], [34], the dolphin's position and velocity cannot be determined simultaneously, so the dolphin's state must be described by wave function $\psi(X, t)$, where X is the dolphin's position vector. The physical meaning of the wave function is that the square of its mode represents the probability density of the dolphin appearing at the X -position in space. The definition of the wave function $\psi(X, t)$ is as follows:

$$|\psi|^2 dx dy dz = Q dx dy dz \quad (25)$$

Among them, Q is a probability density function, which satisfies the following normalization conditions.

$$\int_{-\infty}^{+\infty} |\psi|^2 dx dy dz = \int_{-\infty}^{+\infty} Q dx dy dz = 1 \quad (26)$$

TABLE 3. The configuration of the computer.

Name	Settings
Hardware	
CPU	Intel(R) Core (TM) i7-8550U
Frequency	1.99 GHz
RAM	16.0GB (15.9 GB Available)
Hard drive	1TB
Software	
Operating system	Windows 10
Language	MATLAB R2018a

The dolphin position is obtained by Monte Carlo stochastic simulation, and its update equation is EQUATIONS (27) to (31).

$$P(t) = \theta \cdot P_b(t) + (1 - \theta) \cdot P_g(t) \quad (27)$$

$$mbest(t) = (1/N) \sum_{i=1}^N P_{bi}(t) \quad (28)$$

$$L(t+1) = 2\alpha \cdot |mbest(t) - X(t)| \quad (29)$$

$$\alpha = a - (a - b) \cdot (t/G_{\max}) \quad (30)$$

$$X(t) = P(t) \pm (L/2) \ln(1/u) \quad (31)$$

where P_b and P_g represent the individual and the global optimal position of the population respectively; θ is a random number which obeys uniform distribution on [0,1]; $P(t)$ is the local attraction region of the t th iteration of dolphin, indicating that the position of each dolphin is a random position between the individual optimal position and the global optimal position; $mbest(t)$ is the average value of the individual optimal position of all dolphins in the population; N denotes the size of the population, i.e. the number of dolphins; L denotes the weighted distance between the particle and the average optimal position of the population; u is a random number which obeys uniform distribution on [0,1]; α is called shrinkage-expansion coefficient, which is used to control the convergence rate of dolphins. As iteration progresses, α changes linearly from a to b . Usually $a = 1$, $b = 0.5$, and G_{\max} represents the maximum number of iterations.

The following evolutionary formulas can be obtained from the synthesis formula (27) – (31):

$$X(t+1) = P(t) \pm \alpha |mbest(t) - X(t)| \cdot \ln(1/u) \quad (32)$$

The flow chart of QDSA is shown in FIGURE 1.

IV. RESULTS ANALYSIS AND COMPARISON

A. EXPERIMENTAL PLATFORM, COMPARING ALGORITHMS AND PARAMETER SETTING

In this paper, the main novel work is to propose a new algorithm, namely QDSA, which belongs to the field of artificial intelligence. The performance of computers in this field is very important, which directly affects the calculation results. To test the performance of the QDSA and ensure the fairness of each experiment, we need to ensure that it is carried out on the same computer platform. Based on this common sense, the computer platform configuration is shown in TABLE 3.

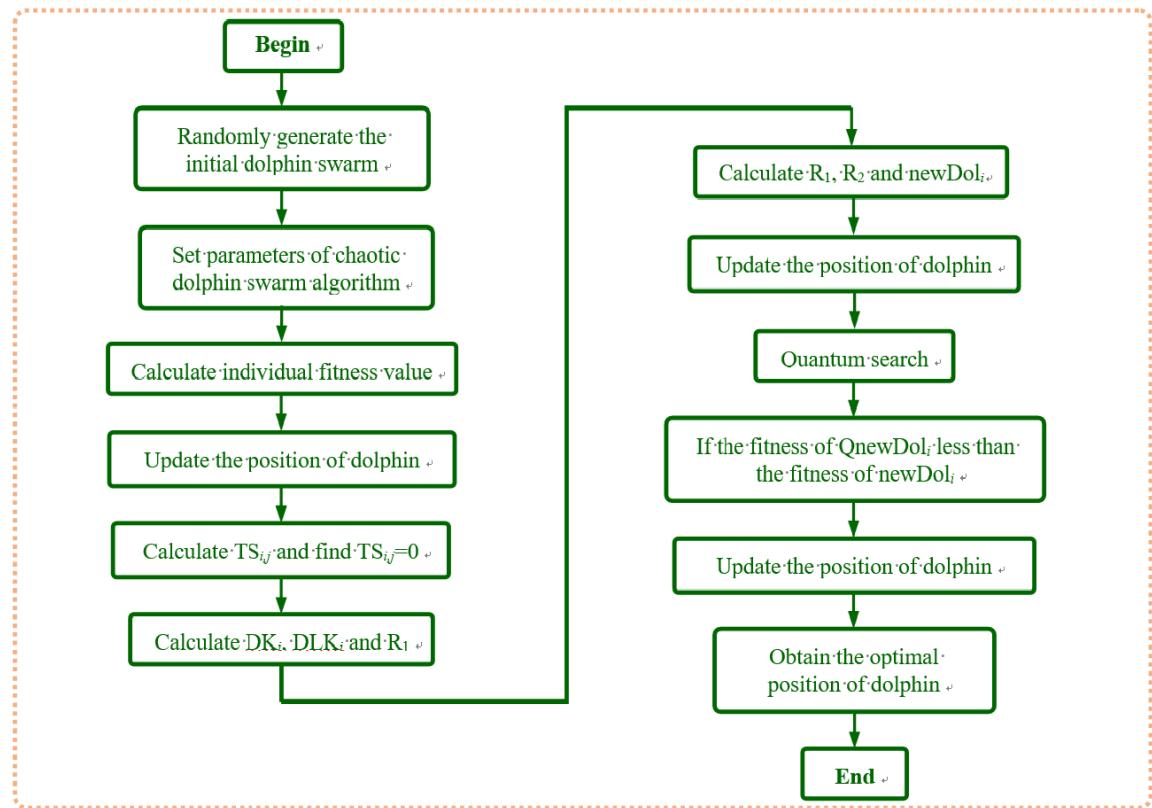


FIGURE 1. The flow chart of QDSA.

For a new algorithm proposed in this paper, it is necessary to test whether the performance of this algorithm is excellent or not and to compare it with the advanced algorithms (e.g. WPA (Wolf pack algorithm)) in the latest research results. In this paper, QDSA is compared with whale optimization algorithm (WOA) [35], DSA ([32]), adaptive genetic algorithm (AGA) [36], adaptive particle swarm optimization (APSO) [22], WPA [30], cuckoo algorithm (CS) [37] and Chicken swarm optimization (CSO) [38].

In this paper, there are many metaheuristic algorithms involved. To ensure that all experiments are carried out on the same platform, the parameters of the algorithm need to be the same. In addition, all experiments are performed 10 times. The average results of these 10 experiments are used as the basis for comparison. The maximum number of iterations is 100 and the population number is 50. Furthermore, to ensure the fairness of the comparison, all the parameters of the proposed and comparative algorithms are set as shown in TABLE 4 in detail.

B. HIGH-DIMENSION TEST FUNCTION

In this paper, to test the performance of the proposed algorithm, several high-dimensional test functions are used, including Rastrigin function, Ackley function, Griewank function, Sphere function, LEVY function, ROTATED HYPER-ELLIPSOID function. In the following research,

TABLE 4. Parameter setting.

Algorithm	Parameter
QDSA	Speed=1, T=10, A=1, e=3
WOA	c1=0.1, c2=0.1, wmax=0.8, wmin=0.1
DSA	Speed=1, T=10, A=1, e=3
AGA	p1_max=0.6, p1_min=0.4, p2_max=0.8, p2_min=0.3
APSO	k1=0.1, k2=0.1, w1x=0.8, w2=0.1
WPA	p=0.05, t=40, step=1, d=4, p_c=0.1;
CS	$\alpha=1$, $\sigma=0.6966$, $\beta=2$, $w=5$
CSO	$R_s=0.6$, $R_h=0.2$, $R_{ch}=0.2$;

the six large-scale test functions are represented by F1, F2, F3, F4, F5, and F6, respectively.

The Rastrigin function has several local minima. It is highly multimodal, but the location of the minimum is regularly distributed. The Ackley function is widely used for testing optimization algorithms. It is characterized by a nearly flat outer region and a large hole at the center. The function poses a risk for optimization algorithms, particularly hill-climbing algorithms, to be trapped in one of its many local minima. The Griewank function has many widespread local minima, which are regularly distributed. The Sphere function has d local minima except for the global one. It is continuous, convex and unimodal. The Rotated Hyper-Ellipsoid function, which is an extension of the axis parallel Hyper-Ellipsoid function. The definitions of these functions are shown in EQUATIONS (33) → (39). Their two-dimensional diagrams are shown in FIGURE 2.

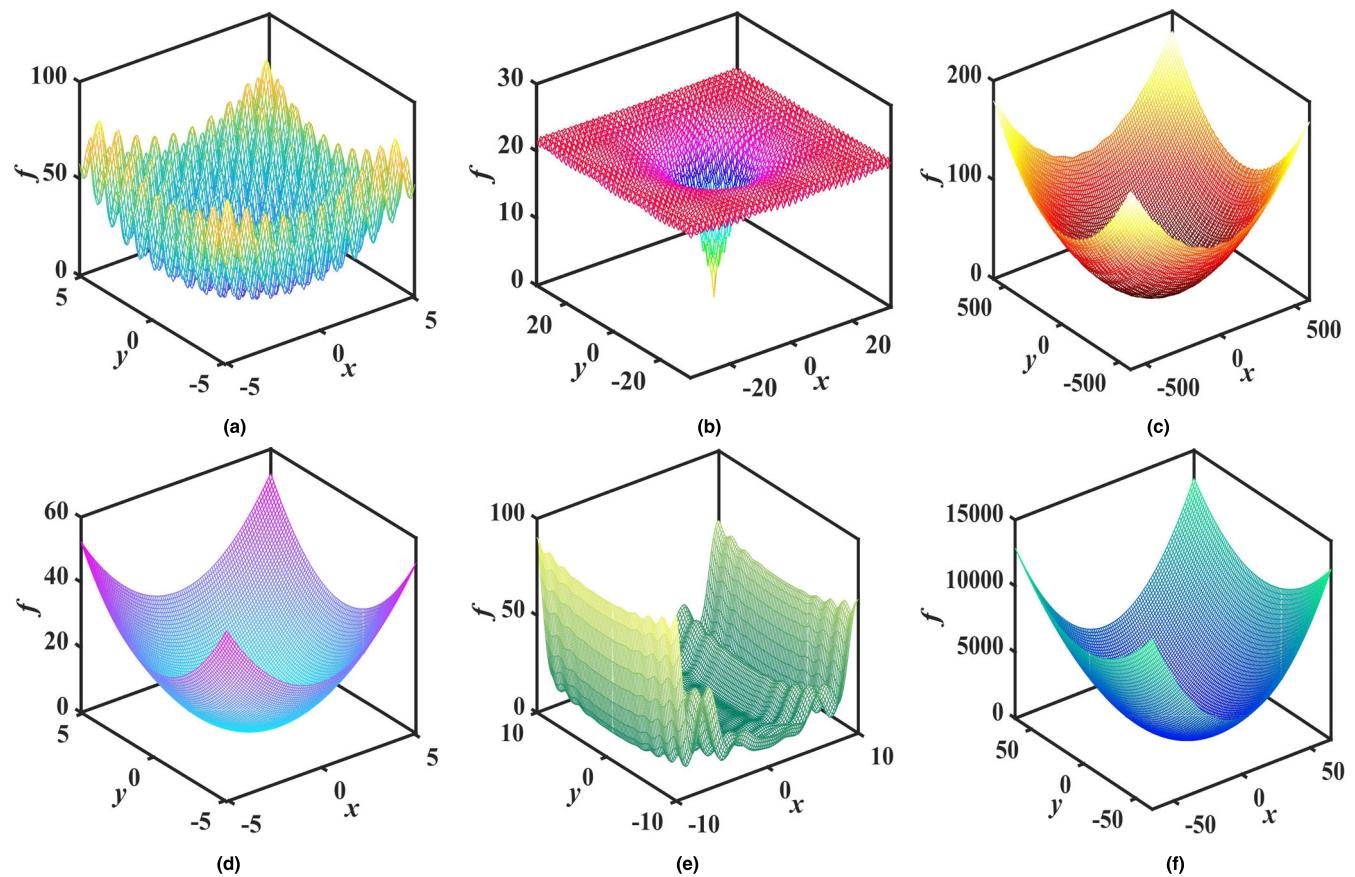


FIGURE 2. Different large-scale test functions: (a) Rastrigin Function; (b) Ackley Function; (c) Griewank Function; (d) Sphere Function; (e) LEVY function; (f) ROTATED HYPER-ELLIPSOID FUNCTION.

The definition of Rastrigin function is as follow:

$$F1(\mathbf{x}) = 10D + \sum_{i=1}^D \left[x_i^2 - 10 \cos(2\pi x_i) \right] \quad (33)$$

where x_i and i belong to $[-5, 5]$ and $[1, D]$, the minimum value of $F1(\mathbf{x})$ is 0.

The definition of Ackley function is as follow:

$$F2(\mathbf{x}) = -a \exp \left(-b \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(cx_i) \right) + a + \exp(1) \quad (34)$$

where x_i and i belong to $[-32.768, 32.768]$ and $[1, D]$; $a = 20$, $b = 0.2$, $c = 2\pi$; the minimum value of $F2(\mathbf{x})$ is 0.

The definition of Griewank function is as follow:

$$F3(\mathbf{x}) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 \quad (35)$$

where x_i and i belong to $[-600, 600]$ and $[1, D]$; the minimum value of $F3(\mathbf{x})$ is 0.

The definition of Sphere function is as follow:

$$F4(\mathbf{x}) = \sum_{i=1}^D x_i^2 \quad (36)$$

where x_i and i belong to $[-5.12, 5.12]$ and $[1, D]$; the minimum value of $F4(\mathbf{x})$ is 0.

The definition of Levy function is as follow:

$$F5(\mathbf{x}) = \sin^2(\pi\omega_1) + \sum_{i=1}^{D-1} (\omega_i - 1)^2 \dots \left[1 + 10 \sin^2(\pi\omega_i + 1) \right] + (\omega_D - 1)^2 \left[1 + \sin^2(2\pi\omega_D) \right] \quad (37)$$

where i belong to $[1, D]$ and the minimum value of $F5(\mathbf{x})$ is 0. ω_i is defined as follow:

$$\omega_i = 1 + \frac{x_i - 1}{4} \quad (38)$$

The definition of Rotated hyper-ellipsoid function is as follow:

$$F6(\mathbf{x}) = \sum_{i=1}^D \sum_{j=1}^i x_j^2 \quad (39)$$

TABLE 5. Different algorithms comparison based on the different dimensions of the F1 using different indicators.

Dimensional number	Indexes	QDSA	WOA	DSA	AGA	APSO	WPA	CS	CSO
200	Min	0.0012	1732.2033	1576.5666	1819.1153	1701.6671	2242.7768	2904.7043	1376.2899
	Max	0.4193	1969.8079	2088.0065	2038.8086	1898.7392	2780.3252	3015.9177	1675.4501
	Mean	0.0857	1873.6345	1836.4739	1936.9587	1835.7118	2588.9182	2968.6296	1497.1214
	Std.	0.1292	63.6635	150.9649	60.9969	55.5443	155.6325	40.3963	99.4531
	A. G	5.27	100.8	91.6	100.7	101.0	67.8	65.7	99.3
300	Min	0.0035	2838.7237	2555.5864	3263.3253	2710.6899	4168.0811	4461.6777	2238.7522
	Max	2.5386	3202.3839	3265.0923	3665.4863	2936.1754	4622.1799	4612.6540	2598.3638
	Mean	0.6129	3000.1531	2794.4272	3465.5719	2832.5370	4405.0464	4535.5601	2353.7239
	Std.	0.9399	125.0540	227.0048	110.7779	61.8723	179.6283	50.3892	113.3802
	A. G	42.6	100.6	90.5	100.7	100.9	76.2	49.3	99.9
400	Min	0.0005	3828.4044	3439.4846	4652.8013	3676.4864	5916.0831	6012.0534	3045.4437
	Max	7.0615	4265.1494	4158.4686	5214.0980	4016.7491	6423.3025	6192.0892	3409.7590
	Mean	0.9925	4052.2951	3753.2094	5003.9845	3819.5730	6247.2088	6135.0815	3256.1027
	Std.	2.1829	138.6597	188.7138	156.8392	112.4930	144.9872	64.4751	131.1519
	A. G	49.4	100.9	92.4	100.7	101.0	67.9	52.7	100.3

Note: The bold value implies the minimum in Min, Max, Mean, Std. and A.G for QDSA and algorithms considered for comparison.

TABLE 6. Different algorithms comparison based on the different dimensions of the F2 using different indicators.

Dimensional number	Indexes	QDSA	WOA	DSA	AGA	APSO	WPA	CS	CSO
200	Min	0.0014	15.1938	12.7378	19.0076	11.9545	19.4989	18.6625	12.0804
	Max	0.2659	16.4528	16.7911	19.7029	12.9804	20.8431	19.2840	13.5799
	Mean	0.0785	15.7938	14.0664	19.2920	12.5256	20.3309	19.0036	12.6813
	Std.	0.0903	0.4038	1.3665	0.2193	0.3366	0.3842	0.2410	0.4337
	A. G	52.3	100.9	98.1	100.8	101.0	87.3	62.3	100.5
300	Min	0.0044	16.1044	12.9695	19.7402	12.2181	20.7582	19.2157	13.0623
	Max	0.3868	16.7625	15.5899	20.1675	13.8656	20.9572	19.4691	14.7597
	Mean	0.1022	16.4225	14.1536	20.0257	12.8487	20.8953	19.3198	13.9809
	Std.	0.1108	0.2144	0.8258	0.1169	0.4217	0.1734	0.1827	0.5331
	A. G	41.9	100.9	92.1	100.5	101.0	58.3	61.0	100.6
400	Min	0.0082	15.8261	13.2463	20.1535	12.0653	20.9190	19.1986	14.2144
	Max	0.3402	16.8868	15.4416	20.4082	13.0286	21.0503	19.6286	14.9610
	Mean	0.1117	16.4417	14.3144	20.3124	12.7232	21.0028	19.4128	14.5740
	Std.	0.1328	0.4046	0.7294	0.1815	0.2939	0.1947	0.1868	0.2574
	A. G	56.4	100.7	97.7	100.6	101.0	60.0	63.7	99.3

Note: The bold value implies the minimum in Min, Max, Mean, Std. and A.G for QDSA and algorithms considered for comparison

where x_i and i belong to $[-65.536, 65.536]$ and $[1, D]$ and the minimum value of $F6(x)$ is 0.

C. EXPERIMENT I: PERFORMANCE TESTING AND COMPARISON WITH ADVANCED ALGORITHMS BASED ON F1

In this experiment, the performance of QDSA and advanced algorithm including WOA, DSA, AGA, APSO, WPA, CS, and CSO is tested based on the high-dimensional function F1. In addition, the five different test indexes (Min (minimum), Max (maximum) Mean (mean), Std. (standard deviation) and A. G (average generation)) are adopted to test the superiority of QDSA. Furthermore, the dimensions of F1 function are set to 200, 300 and 400 respectively. The parameters of the algorithm in this subsection whose detailed parameters are set in TABLE 4 are specified in Section IV.A.

Some experimental results are shown in FIGURES 3-4 and TABLE 5. FIGURE 3 gives the iterative behavior based on

QDSA using the different dimension of the F1. FIGURE 4 shows the convergence curve based on QDSA and advanced algorithms considered for comparison using the different dimension of the F1. TABLE 5 lists different algorithms comparison based on the different dimensions of the F1 using different indicators.

By analyzing FIGURES 3-4 and TABLE 5, the following results and comparison are given.

(1) As can be seen from FIGURE 3 (a), (d) and (g) with the increasing number of iterations, the average fitting value gradually approaches the optimal solution of F1 function, i.e. 0. This situation is also reflected in FIGURE 3 (b), (c), (e), (f), (h) and (i). This result shows that QDSA can obtain the optimal solution of F1.

(2) For FIGURE 4, the average fitting curve of QDSA is lower than that of other advanced algorithms, which indicates that QDSA converges more easily than other algorithms. Furthermore, with the increase of the dimension of F1 function, the range of solution is also gradually enlarged, but the

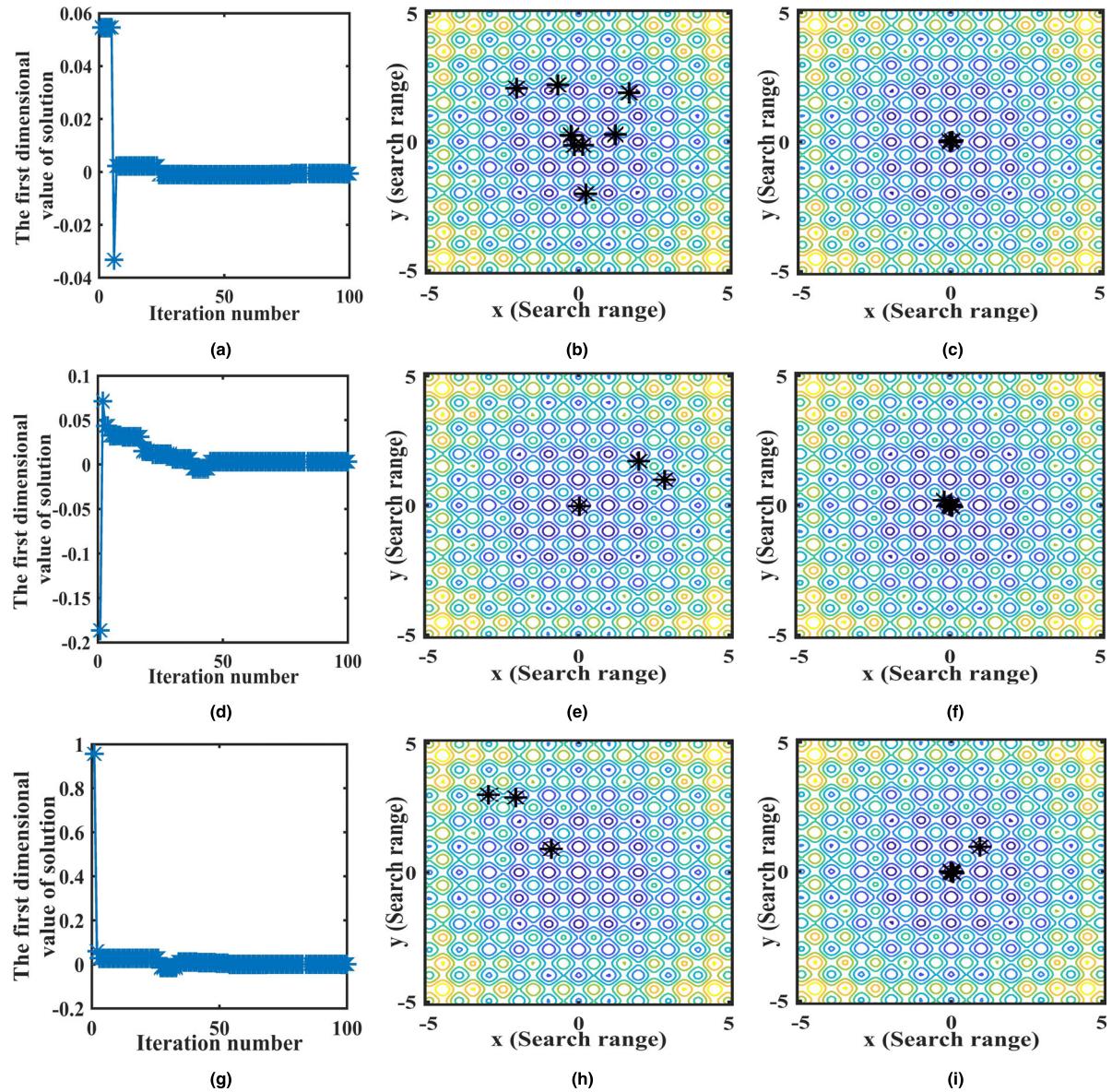


FIGURE 3. Iterative behavior based on QDSA using the different dimension of the F1: The variation of the first dimensional solution value in view of 200,300,400-dimension consisting of (a), (d) and (g); Search range of the first individual in the population in view of 200,300,400-dimension including (b), (e) and (h); Search range for the optimal individual in the population in view of 200,300,400-dimension including (c), (f) and (i).

average fitting curve of QDSA is still very low, which shows that QDSA has a strong convergence ability.

(3) From TABLE 5, the Min, Max, Mean, Std., and A.G of QDSA are lower than other algorithms based on the 200, 300 and 400 dimensions of F1. This result shows that QDSA is better than other algorithms.

D. EXPERIMENT II: PERFORMANCE TESTING AND COMPARISON WITH ADVANCED ALGORITHMS BASED ON F2

Some experimental results are expressed in FIGUREs 5-6 and TABLE 6. FIGURE 5 shows the iterative behavior based on QDSA using the different dimension of the F2 including

200, 300, and 400 dimensions. FIGURE 6 demonstrates the convergence curve based on QDSA and some advanced algorithms considered for comparison by applying the different dimension of the F2. TABLE 6 lists different algorithms comparison based on the different dimensions of the F2 using different indicators.

Based on the FIGUREs 5-6 and TABLE 6, the following results and comparison are given.

(1) As can be seen from FIGURE 5 (a), (d) and (g) with the increasing number of iterations, the average fitting value gradually approaches the optimal solution of F2 function, i.e. 0. This situation is also reflected in

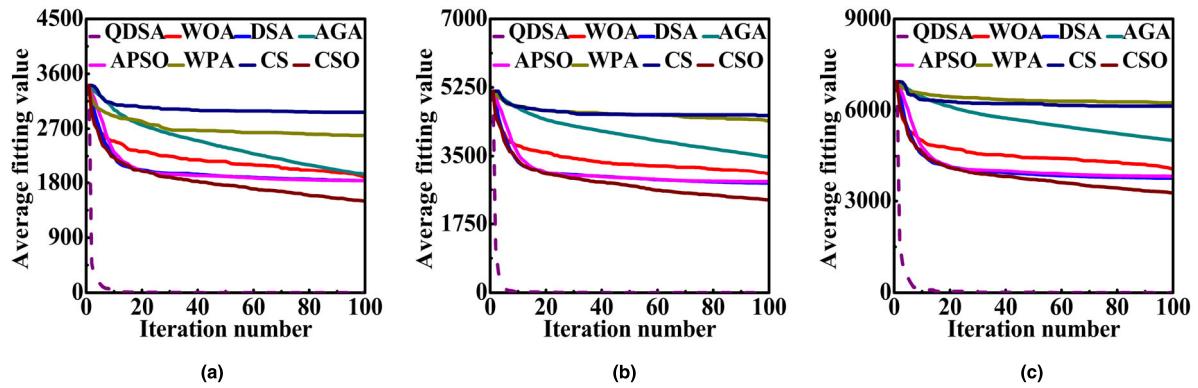


FIGURE 4. Convergence curve based on QDSA and advanced algorithms considered for comparison using the different dimension of the F1: (a) 200 dimension; (b) 300 dimension; (c) 400 dimension.

TABLE 7. Different algorithms comparison based on the different dimensions of the F3 using different indicators.

Dimensional number	Indexes	QDSA	WOA	DSA	AGA	APSO	WPA	CS	CSO
200	Min	0.0050	465.4046	178.8862	1507.8385	192.2975	2240.7255	353.9997	350.0890
	Max	0.5430	744.1727	278.5659	2100.0928	322.6654	4900.3287	496.2843	520.2785
	Mean	0.2083	567.1651	223.6528	1812.8860	235.7809	3112.4005	410.8370	421.4664
	Std.	0.1763	90.7791	30.2853	145.3589	35.0199	782.4435	47.9413	55.3516
300	A. G	75.1	100.4	100.6	100.9	101.0	99.0	85.9	100.6
	Min	0.0008	970.6656	286.1195	3716.2391	340.1464	5916.5173	619.4036	624.6368
	Max	1.2005	1162.7575	398.2741	4259.9276	412.7989	7576.0909	798.2705	920.1294
	Mean	0.2755	1047.0260	343.2535	3892.9801	372.7904	7257.0004	683.2566	760.3473
400	Std.	0.3982	64.6768	33.0778	195.0262	23.7173	503.7684	60.5275	95.0334
	A. G	52.2	100.9	100.8	100.7	101.0	101.0	56.4	100.6
	Min	0.0020	1129.0692	448.6215	5391.1855	488.2896	9533.2193	852.4480	958.0114
	Max	1.0082	1593.0141	560.1625	6358.4434	604.0710	10452.8967	1184.3434	1198.6750
400	Mean	0.2080	1420.8865	500.6400	5809.9640	531.3723	10011.1381	1055.9470	1099.3760
	Std.	0.3245	142.6549	35.0174	317.2044	35.2934	277.4350	90.7366	71.9146
	A. G	43.7	100.9	100.1	100.7	101.0	101.0	64.1	101.0

Note: The bold value implies the minimum in Min, Max, Mean, Std. and A.G for QDSA and algorithms considered for comparison.

FIGURE 5 (b), (c), (e), (f), (h) and (i). This implies that QDSA can obtain the optimal solution of F2.

(2) For FIGURE 6, the average fitting curve of QDSA is lower than that of other algorithms, which implies that QDSA converges more easily than other algorithms. In addition, with the increase of the dimension of F2 function, the range of solution is also gradually enlarged, however, compared with other algorithms, QDSA has very good convergence ability.

(3) From TABLE 6, the Min, Max, Mean, Std., and A.G of QDSA are $11.9531 \rightarrow 19.4975$, $12.7145 \rightarrow 20.5772$, $12.4471 \rightarrow 20.2524$, $0.1290 \rightarrow 1.2762$ and $10 \rightarrow 48.7$ lower than other algorithms based on the 200 dimensions of F2. Furthermore, the Min, Max, Mean, Std., and A.G of QDSA are $12.2137 \rightarrow 20.7538$, $13.4788 \rightarrow 20.5704$, $12.7465 \rightarrow 20.7931$, $0.0061 \rightarrow 0.7150$ and $16.4 \rightarrow 59.1$ lower than other algorithms based on the 300 dimensions of F2. Furthermore, the Min, Max, Mean, Std., and A.G of QDSA are $12.0571 \rightarrow 20.9108$, $12.6884 \rightarrow 20.7101$, $12.6115 \rightarrow 20.8911$, $0.0487 \rightarrow 0.5966$ and $3.6 \rightarrow 44.6$ lower than other algorithms based on the 300 dimensions of F2. These results show that QDSA is better than other algorithms.

E. EXPERIMENT III: PERFORMANCE TESTING AND COMPARISON WITH ADVANCED ALGORITHMS BASED ON F3

Some important results are demonstrated in FIGURES 7-8 and TABLE 7. FIGURE 7 lists the iterative behavior based on QDSA using the different dimension of the F3 including 200, 300, and 400 dimensions. FIGURE 8 shows the convergence curve based on QDSA and advanced algorithms considered for comparison using the different dimension of the F3. TABLE 7 lists different algorithms comparison based on the different dimensions of the F3 using different indicators.

Based on the FIGURES 7-8 and TABLE 7, the following results and comparison are given.

(1) As can be seen from FIGURE 7 (a), (d) and (g) with the increasing number of iterations, the average fitting value gradually approaches the optimal solution of F3 function, i.e. 0. This situation is also reflected in FIGURE 7 (b), (c), (e), (f), (h) and (i). This shows that QDSA can obtain the optimal solution of F3.

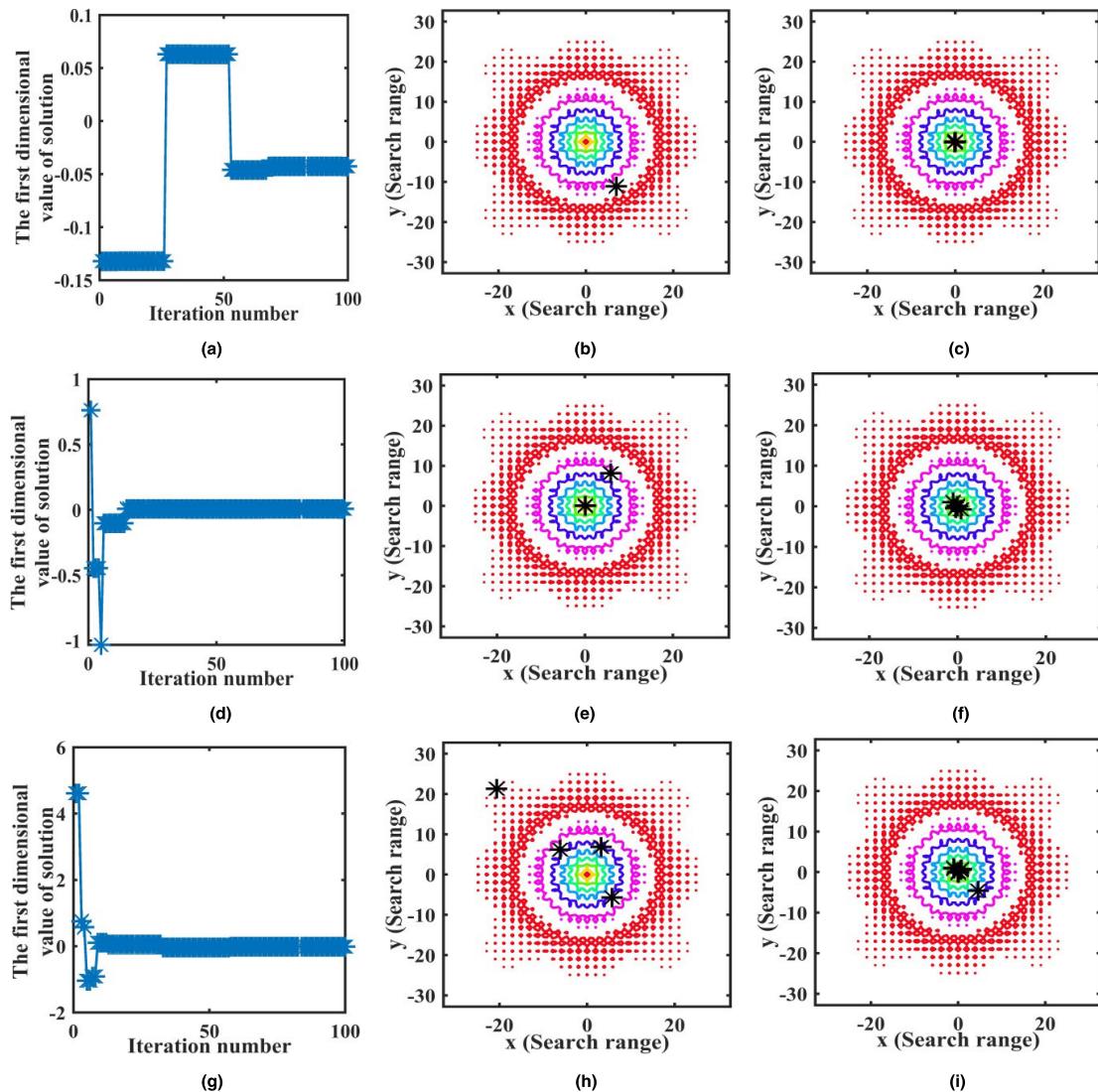


FIGURE 5. Iterative behavior based on QDSA using the different dimension of the F2: The variation of the first dimensional solution value in view of 200,300,400-dimension consisting of (a), (d) and (g); Search range of the first individual in the population in view of 200,300,400-dimension including (b), (e) and (h); Search range for the optimal individual in the population in view of 200,300,400-dimension including (c), (f) and (i).

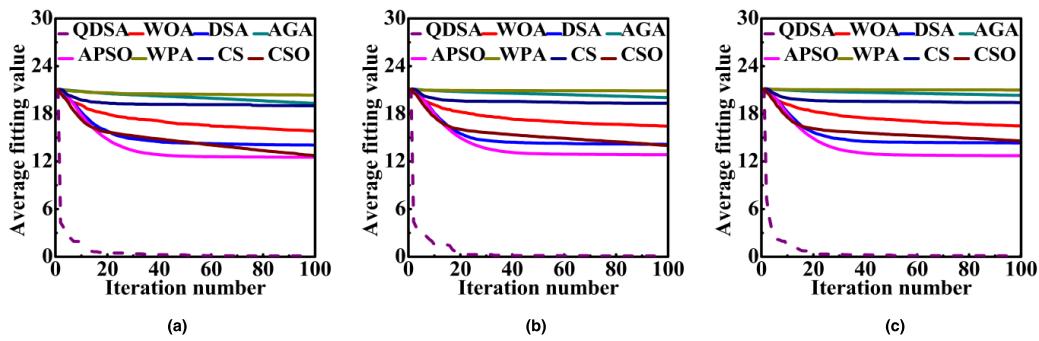


FIGURE 6. Convergence curve based on QDSA and advanced algorithms considered for comparison using the different dimension of the F2: (a) 200 dimension; (b) 300 dimension; (c) 400 dimension.

(2) For FIGURE 8, the average fitting curve of QDSA is lower than that of other algorithms, which implies that QDSA converges more easily than other algorithms. Besides, with

the increase of the dimension of F3 function, the range of solution is also gradually enlarged, however, compared with other algorithms, QDSA has very good convergence ability.

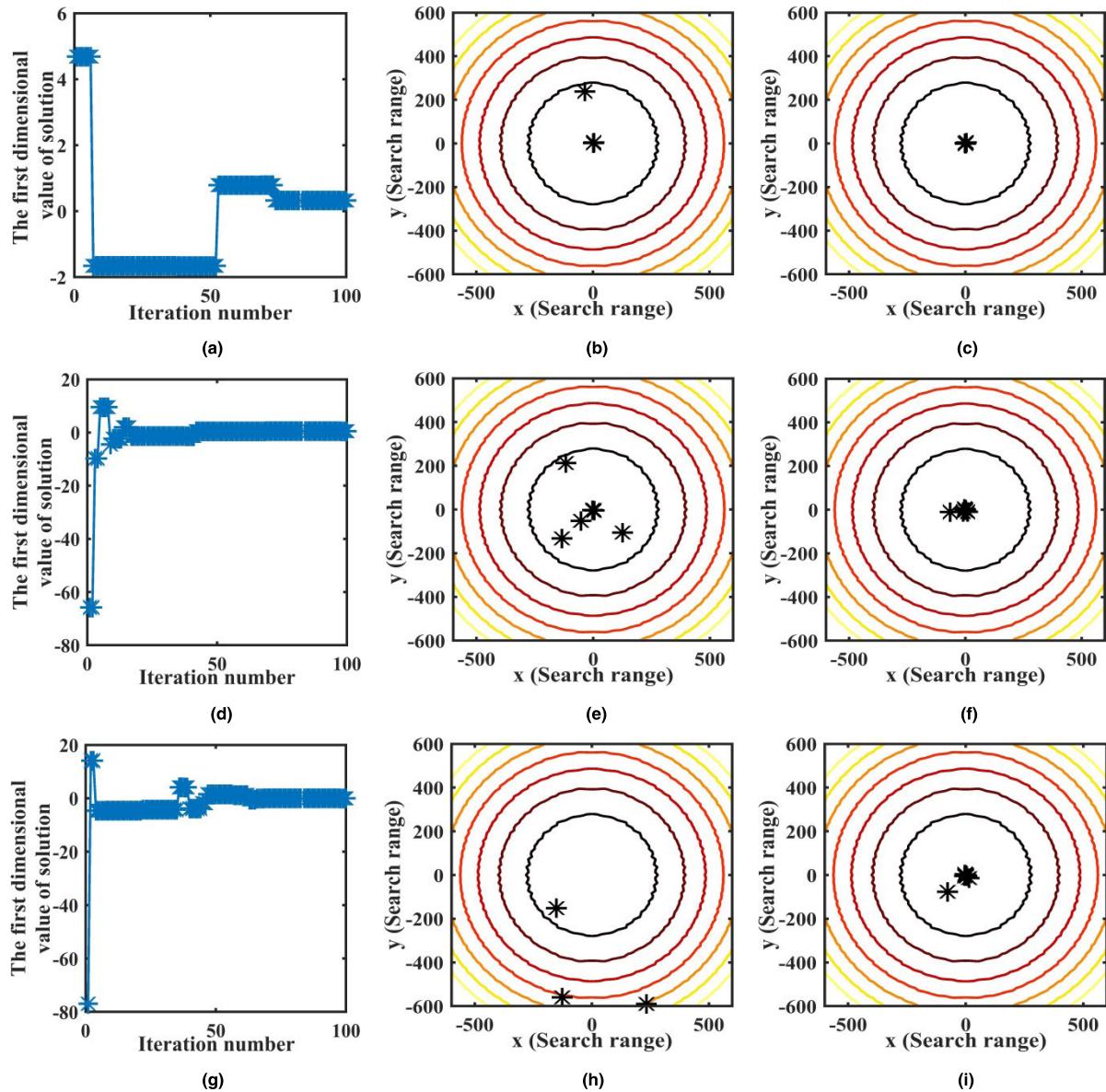


FIGURE 7. Iterative behavior based on QDSA using the different dimension of the F3: The variation of the first dimensional solution value in view of 200,300,400-dimension consisting of (a), (d) and (g); Search range of the first individual in the population in view of 200,300,400-dimension including (b), (e) and (h); Search range for the optimal individual in the population in view of 200,300,400-dimension including (c), (f) and (i).

(3) From TABLE 7, the Min, Max, Mean, Std., and A.G of QDSA are $178.8812 \rightarrow 2240.7210$, $278.0229 \rightarrow 4899.7860$, $223.4445 \rightarrow 3112.1920$, $30.1090 \rightarrow 782.2672$ and $10.8 \rightarrow 25.9$ lower than other algorithms based on the 200 dimensions of F3. Moreover, the Min, Max, Mean, Std., and A.G of QDSA are $286.1187 \rightarrow 5916.5170$, $397.0736 \rightarrow 7574.8900$, $342.9780 \rightarrow 7256.7250$, $23.3191 \rightarrow 503.3702$ and $4.2 \rightarrow 48.8$ lower than other algorithms based on the 300 dimensions of F3. Furthermore, the Min, Max, Mean, Std., and A.G of QDSA are $448.6195 \rightarrow 9533.2170$, $559.1543 \rightarrow 10451.8900$, $500.4320 \rightarrow 10010.93$, $34.6929 \rightarrow 316.8799$ and $20.4 \rightarrow 57.3$ lower than other algorithms based

on the 300 dimensions of F3. These results indicate that QDSA is better than other algorithms.

F. EXPERIMENT IV: PERFORMANCE TESTING AND COMPARISON WITH ADVANCED ALGORITHMS BASED ON F4

Some experimental results are shown in FIGUREs 9-10 and TABLE 8. FIGURE 9 gives the Iterative behavior based on QDSA using the different dimensions of the F4 including 200, 300, and 400 dimensions. FIGURE 10 shows the convergence curve based on QDSA and advanced algorithms

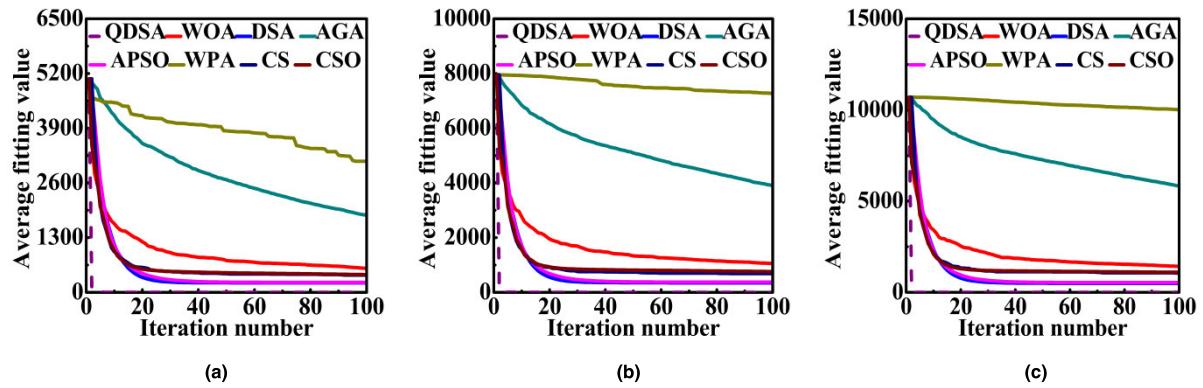


FIGURE 8. Convergence curve based on QDSA and advanced algorithms considered for comparison using the different dimension of the F3: (a) 200 dimension; (b) 300 dimension; (c) 400 dimension.

TABLE 8. Different algorithms comparison based on the different dimensions of the F4 using different indicators.

Dimensional number	Indexes	QDSA	WOA	DSA	AGA	APSO	WPA	CS	CSO
200	Min	0.0000	140.5862	53.0703	456.1761	70.4043	268.3447	1149.7252	78.5121
	Max	0.0004	178.9622	78.8676	587.6073	110.1463	372.6615	1298.2327	124.0715
	Mean	0.0001	159.2023	65.0611	511.3094	85.0944	317.1215	1224.4703	100.9617
	Std.	0.0001	12.1070	8.6077	37.2165	14.0634	28.5008	48.3979	13.5691
300	A. G	47.1	100.7	99.5	100.8	101.0	61.0	47.7	100.3
	Min	0.0000	254.5471	84.6533	1065.1095	118.5716	513.5000	1898.4251	151.1796
	Max	0.0115	363.0833	123.4323	1205.3629	142.0723	707.6119	2028.2575	192.8963
	Mean	0.0032	301.2574	108.5612	1124.3031	131.0120	619.2143	1969.0149	173.0873
400	Std.	0.0037	31.8187	10.8346	42.0474	6.9417	68.3872	42.2973	14.3359
	A. G	46.3	100.5	99.1	100.8	101.0	52.9	51.5	100.3
	Min	0.0000	353.7871	143.1615	1649.2108	170.3074	811.9920	2579.7598	217.0978
	Max	0.0065	487.8017	183.3725	1874.5312	211.5640	1137.3719	2847.8990	267.2438
400	Mean	0.0019	413.6943	158.3961	1769.0476	188.9916	906.8528	2715.2591	240.0333
	Std.	0.0024	44.5609	13.6019	70.4416	12.4001	99.0370	102.8902	17.5562
	A. G	38.0	100.7	96.4	100.8	101.0	87.5	49.8	100.4

Note: The bold value implies the minimum in Min, Max, Mean, Std. and A.G for QDSA and algorithms considered for comparison.

TABLE 9. Different algorithms comparison based on the different dimensions of the F5 using different indicators.

Dimensional number	Indexes	QDSA	WOA	DSA	AGA	APSO	WPA	CS	CSO
200	Min	18.6086	201.8593	140.3948	528.6778	143.1933	469.8905	1263.2069	103.7267
	Max	18.6634	321.4239	245.4440	688.6157	227.4795	1071.8616	1649.9686	168.2907
	Mean	18.6459	245.1045	198.6244	627.0805	185.3753	792.8890	1435.6080	131.3074
	Std.	0.0178	36.6945	42.1038	58.1022	31.9880	208.5624	117.4375	19.8573
300	A. G	46.9	100.7	100.6	101.0	101.0	82.1	55.3	99.4
	Min	27.7185	288.7524	250.1760	1120.9896	240.4186	1559.0298	2132.5616	245.6652
	Max	27.7629	449.7682	386.1008	1409.8434	354.8639	2512.2295	2450.0570	348.3573
	Mean	27.7435	374.3485	293.3272	1255.4652	282.1136	2117.6702	2319.4728	297.7489
400	Std.	0.0161	52.5551	35.6248	90.5394	39.4388	330.1524	130.1980	37.3296
	A. G	49.8	100.6	100.8	100.8	101.0	84.3	60.7	98.9
	Min	36.8248	414.5790	339.1727	1947.3718	330.8262	3190.1033	2666.8663	382.6486
	Max	36.8470	742.4183	667.7638	2296.8867	429.9525	3628.0937	3632.3179	518.3401
400	Mean	36.8373	549.1826	456.4137	2090.0741	389.2274	3454.4766	3314.6756	430.1580
	Std.	0.0075	106.3903	95.6657	117.8666	31.7891	124.5031	274.5527	41.9025
	A. G	37.1	100.7	100.4	100.9	101.0	86.2	42.8	99.6

Note: The bold value implies the minimum in Min, Max, Mean, Std. and A.G for QDSA and algorithms considered for comparison.

considered for comparison using the different dimensions of the F4. TABLE 8 lists different algorithms comparison based on the different dimensions of the F4 using different indicators.

Based on the FIGURES 9-10 and TABLE 8, the following results and comparison are given.

(1) As can be seen from FIGURE 9 (a), (d) and (g) with the increasing number of iterations, the average fitting value gradually approaches the optimal solution of F4 function, i.e. 0. This situation is also reflected in FIGURE 9 (b), (c), (e), (f), (h) and (i). This shows that QDSA can obtain the optimal solution of F4.

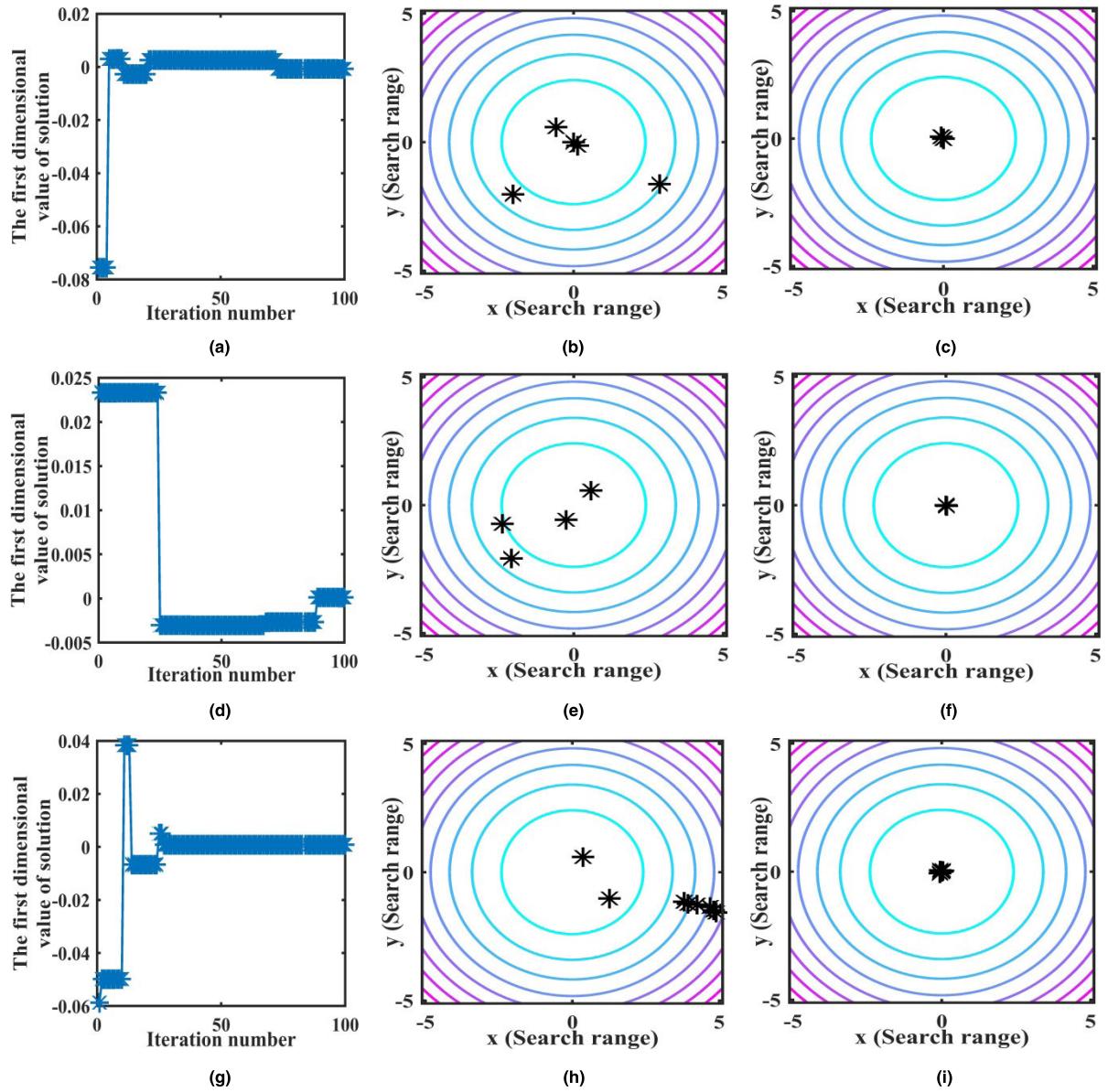


FIGURE 9. Iterative behavior based on QDSA using the different dimension of the F4: The variation of the first dimensional solution value in view of 200,300,400-dimension consisting of (a), (d) and (g); Search range of the first individual in the population in view of 200,300,400-dimension including (b), (e) and (h); Search range for the optimal individual in the population in view of 200,300,400-dimension including (c), (f) and (i).

(2) For FIGURE 9, the average fitting curve of QDSA is lower than that of other algorithms, which implies that QDSA converges more easily than other algorithms. Besides, with the increase of the dimension of F4 function, the range of solution is also gradually enlarged, however, compared with other algorithms, QDSA has very good convergence ability.

(3) From TABLE 8, the Min, Max, Mean, Std., and A.G of QDSA are $53.0703 \rightarrow 1149.7250$, $78.8672 \rightarrow 1298.2320$, $65.0610 \rightarrow 1224.4700$, $8.6076 \rightarrow 48.3978$ and $0.6 \rightarrow 53.9$ lower than other algorithms based on the 200 dimensions of F4. In addition, the Min, Max, Mean, Std., and A.G of QDSA are $84.6533 \rightarrow 1898.4250$, $123.4208 \rightarrow 2028.246$, $108.5580 \rightarrow 1969.0120$, $6.9380 \rightarrow 68.3835$ and $5.2 \rightarrow 54.7$

lower than other algorithms based on the 300 dimensions of F4. Furthermore, the Min, Max, Mean, Std., and A.G of QDSA are $143.1615 \rightarrow 2597.7600$, $183.3660 \rightarrow 2847.8930$, $158.3942 \rightarrow 2715.2570$, $12.3977 \rightarrow 102.8878$ and $11.8 \rightarrow 63$ lower than other algorithms based on the 300 dimensions of F4. These results indicate that QDSA is better than other algorithms.

G. EXPERIMENT V: PERFORMANCE TESTING AND COMPARISON WITH ADVANCED ALGORITHMS BASED ON F5

Some important results are shown in FIGURES 11-12 and TABLE 9. FIGURE 11 gives the iterative behavior based on

TABLE 10. Different algorithms comparison based on the different dimensions of the F6 using different indicators.

Dimensional number	Indexes	QDSA	WOA	DSA	AGA	APSO	WPA	CS	CSO
200	Min	0.0025	1923675	722954.4	7257614	889372.5	7422382	3884542	1005488
	Max	283.5402	3573878	1166612	9653726	1315803	15492990	5749109	1631351
	Mean	35.9024	2816644	1058252	8511351	1087769	9752493	5009613	1224052
	Std.	87.2713	510884.6	131978.4	792571.1	115415.5	2230245	746840	193382.8
	A. G	51.5	100.8	101.0	100.9	101.0	101.0	58.0	100.7
300	Min	0.2149	6173820	1850158	23825594	2286022	24715128	12164279	2742590
	Max	1953.8998	7377421	2748650	29234314	3951869	46756316	15357399	4321004
	Mean	206.0836	6764807	2345509	26033493	2716361	37720847	13816065	3536834
	Std.	614.2517	394000.4	318314.7	1604732	530028.6	8011242	860308.3	551010
	A. G	50.6	100.8	101.0	100.6	101.0	101.0	57.9	100.7
400	Min	0.9470	11473645	4117686	52512917	4478208	65827861	24065831	5423230
	Max	308.1325	16218312	5156037	59965463	5310980	81677495	29252532	8051840
	Mean	74.9902	13464947	4663891	56208686	5010510	77610697	26066818	6810977
	Std.	108.5540	1414623	311992.2	2542145	251384.6	4986482	1469698	986547.4
	A. G	47.8	100.4	100.8	100.9	101.0	101.0	57.5	100.6

Note: The bold value implies the minimum in Min, Max, Mean, Std. and A.G for QDSA and algorithms considered for comparison.

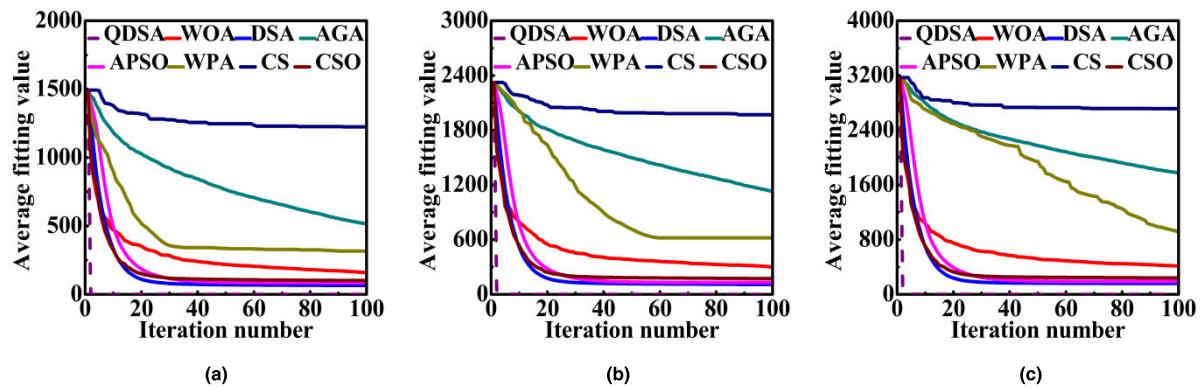


FIGURE 10. Convergence curve based on QDSA and advanced algorithms considered for comparison using the different dimension of the F4: (a) 200 dimension; (b) 300 dimension; (c) 400 dimension.

QDSA using the different dimension of the F5 including 200, 300, and 400 dimensions. FIGURE 12 shows the convergence curve based on QDSA and advanced algorithms considered for comparison using the different dimension of the F5. TABLE 9 lists different algorithms comparison based on the different dimensions of the F5 using different indicators.

Based on the FIGUREs 11-12 and TABLE 9, the following results and comparison are given.

(1) As can be seen from FIGURE 11 (a), (d) and (g) with the increasing number of iterations, the average fitting value gradually approaches the optimal solution of F5 function, i.e. 0. This situation is also reflected in FIGURE 11 (b), (c), (e), (f), (h) and (i). This shows that QDSA can obtain the optimal solution of F5.

(2) For FIGURE 11, the average fitting curve of QDSA is lower than that of other algorithms, which implies that QDSA converges more easily than other algorithms. What is more, with the increase of the dimension of F5 function, the range of solution is also gradually enlarged, however, compared with other algorithms, QDSA has very good convergence ability.

(3) From TABLE 9, the Min, Max, Mean, Std., and A.G of QDSA are $85.1181 \rightarrow 1240.5980$, $149.6273 \rightarrow 1631.3050$, $112.6615 \rightarrow 1416.9620$, $19.8395 \rightarrow 208.5446$ and $8.4 \rightarrow 54.1$ lower than other algorithms based on the 200 dimensions of F5. In addition, the Min, Max, Mean, Std., and A.G of QDSA are $212.7001 \rightarrow 2104.8430$, $320.5944 \rightarrow 2484.4670$, $254.3701 \rightarrow 2291.7290$, $35.6087 \rightarrow 330.1363$ and $10.9 \rightarrow 51.2$ lower than other algorithms based on the 300 dimensions of F5. Furthermore, the Min, Max, Mean, Std., and A.G of QDSA are $294.0014 \rightarrow 3153.2790$, $393.1055 \rightarrow 3595.4710$, $352.3901 \rightarrow 3417.6390$, $31.7816 \rightarrow 274.5452$ and $5.7 \rightarrow 63.9$ lower than other algorithms based on the 300 dimensions of F5. These results indicate that QDSA is better than other algorithms.

H. EXPERIMENT VI: PERFORMANCE TESTING AND COMPARISON WITH ADVANCED ALGORITHMS BASED ON F6

Some important results obtained by this experiment are shown in FIGUREs 13-14 and TABLE 10. FIGURE 13 gives

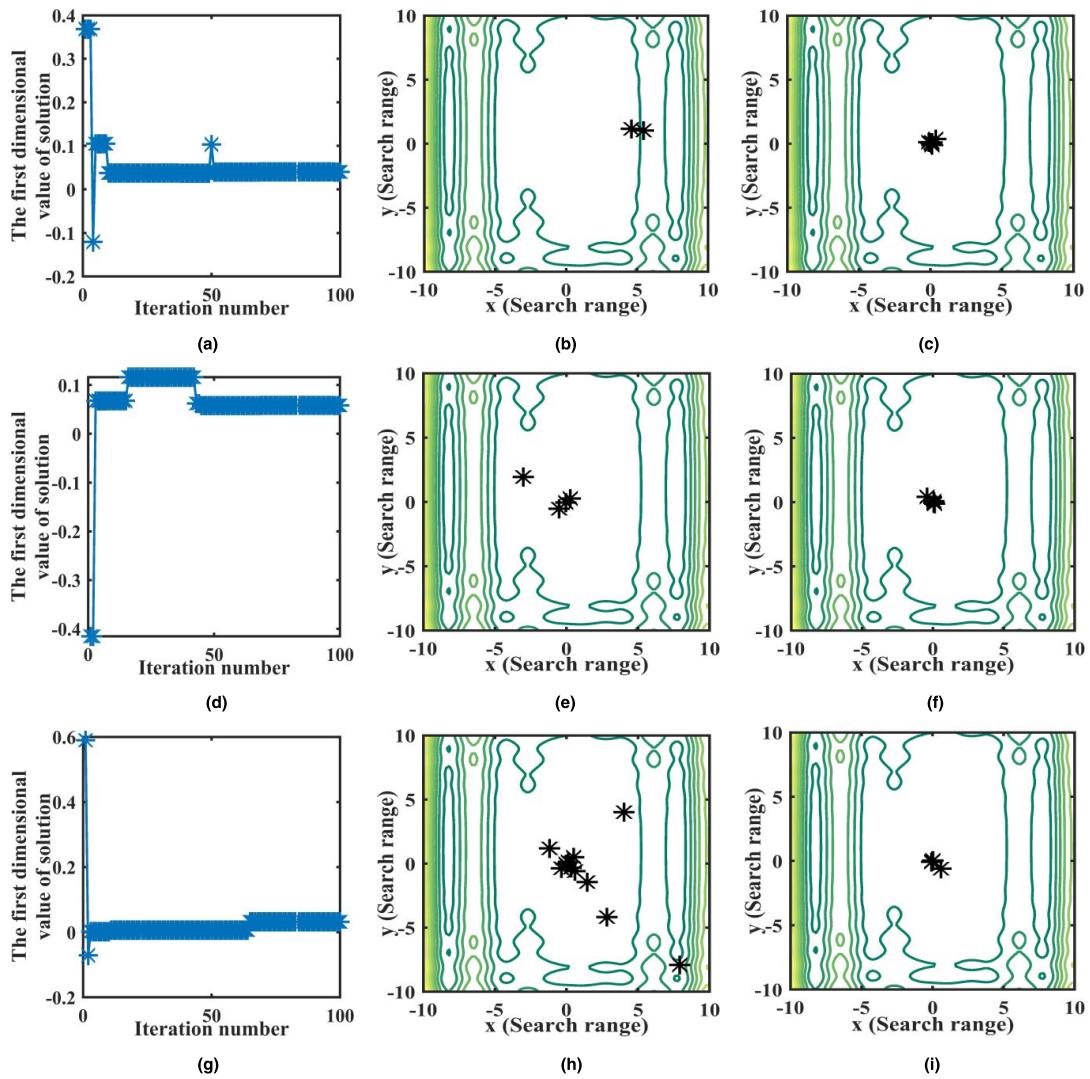


FIGURE 11. Iterative behavior based on QDSA using the different dimension of the F5: The variation of the first dimensional solution value in view of 200,300,400-dimension consisting of (a), (d) and (g); Search range of the first individual in the population in view of 200,300,400-dimension including (b), (e) and (h); Search range for the optimal individual in the population in view of 200,300,400-dimension including (c), (f) and (i).

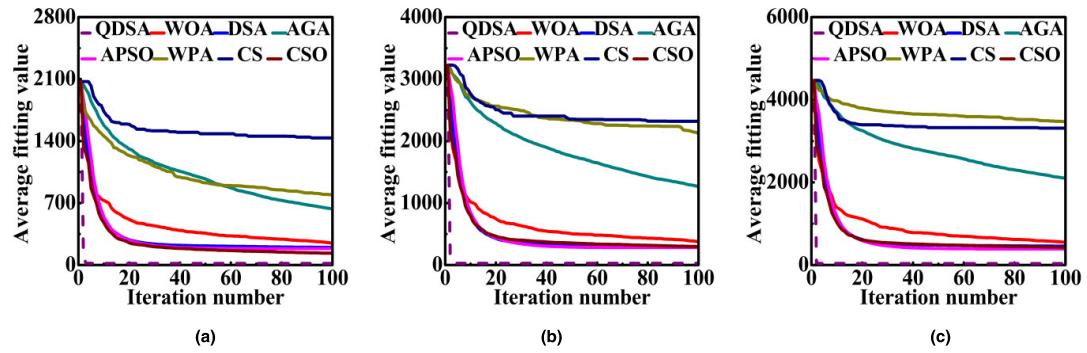


FIGURE 12. Convergence curve based on QDSA and advanced algorithms considered for comparison using the different dimension of the F6: (a) 200 dimension; (b) 300 dimension; (c) 400 dimension.

the iterative behavior based on QDSA using the different dimensions of the F6. FIGURE 14 shows the convergence curve based on QDSA and advanced algorithms considered

for comparison using the different dimensions of the F6. TABLE 10 lists different algorithms comparison based on the different dimensions of the F6 using different indicators.

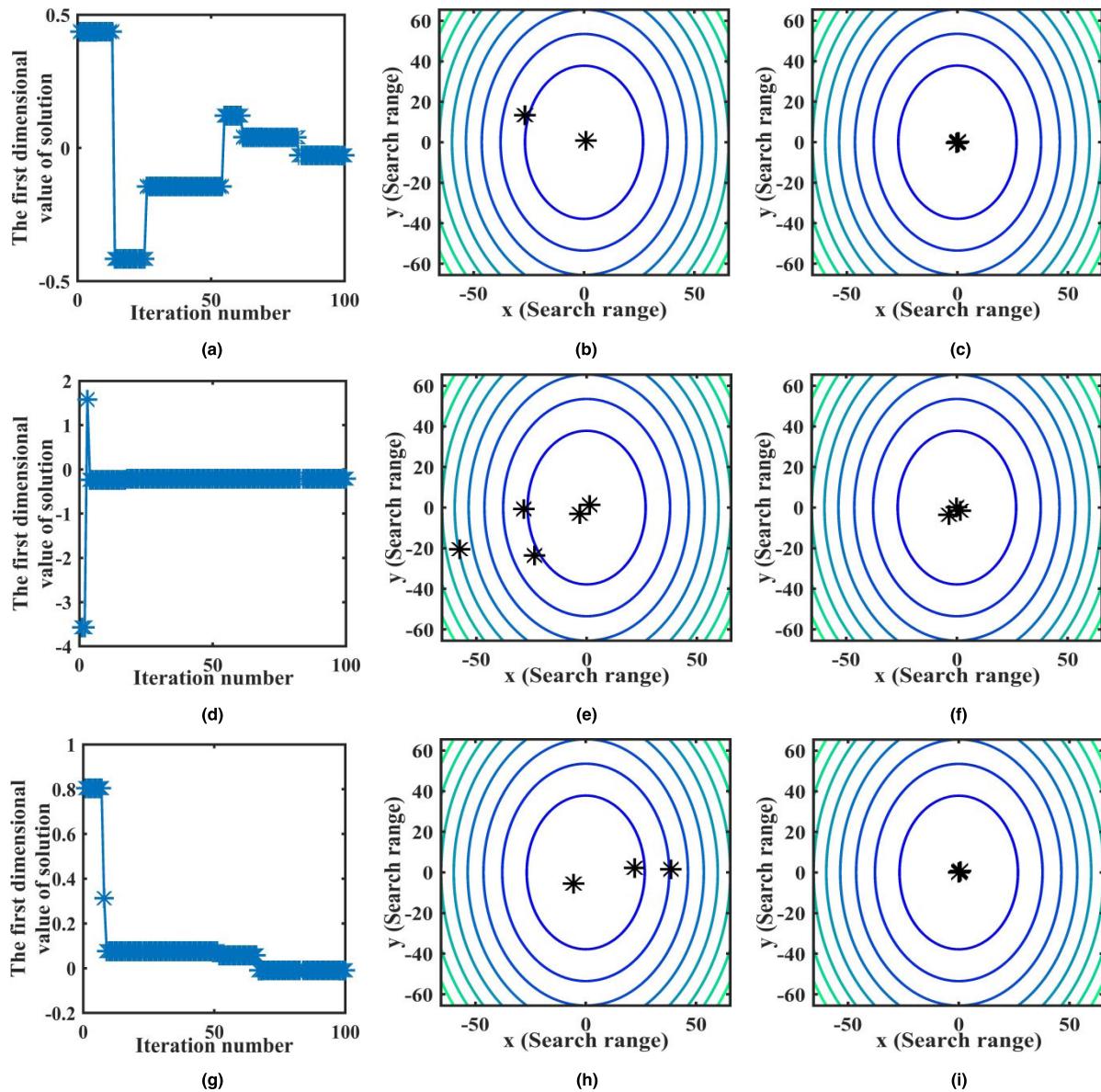


FIGURE 13. Iterative behavior based on QDSA using the different dimension of the F6: The variation of the first dimensional solution value in view of 200,300,400-dimension consisting of (a), (d) and (g); Search range of the first individual in the population in view of 200,300,400-dimension including (b), (e) and (h); Search range for the optimal individual in the population in view of 200,300,400-dimension including (c), (f) and (i).

Based on the FIGURES 13-14 and TABLE 10, the following results and comparison are given.

(1) As can be seen from FIGURE 13 (a), (d) and (g) with the increasing number of iterations, the average fitting value gradually approaches the optimal solution of F6 function, i.e. 0. This situation is also reflected in FIGURE 13 (b), (c), (e), (f), (h) and (i). This shows that QDSA can obtain the optimal solution of F6. Furthermore, with the increase of the dimension of F6 function, the range of solution is also gradually enlarged, however, compared with other algorithms, QDSA has very good convergence ability.

(2) For FIGURE 14, the average fitting curve of QDSA is lower than that of other algorithms, which implies that QDSA converges more easily than other algorithms.

(3) From TABLE 10, the Min, Max, Mean, Std., and A.G of QDSA are $722954.4 \rightarrow 7422382$, $1166328 \rightarrow 15492706$, $1058216 \rightarrow 9752457$, $115328.2 \rightarrow 2230158$ and $6.5 \rightarrow 4.9.5$ lower than other algorithms based on the 200 dimensions of F6. In addition, the Min, Max, Mean, Std., and A.G of QDSA are $1850158 \rightarrow 24715128$, $2746696 \rightarrow 46754362$, $2345303 \rightarrow 37720641$, $317700.4 \rightarrow 8010628$ and $7.3 \rightarrow 50.4$ lower than other algorithms based on the 300 dimensions of F6. Furthermore, the Min, Max, Mean, Std., and A.G of QDSA are $4117685 \rightarrow 65827860$, $5155729 \rightarrow 81677187$, $4663816 \rightarrow 77610622$, $251276 \rightarrow 4986373$ and $9.7 \rightarrow 53.2$ lower than other algorithms based on the 300 dimensions of F6. These results indicate that QDSA is better than other algorithms.

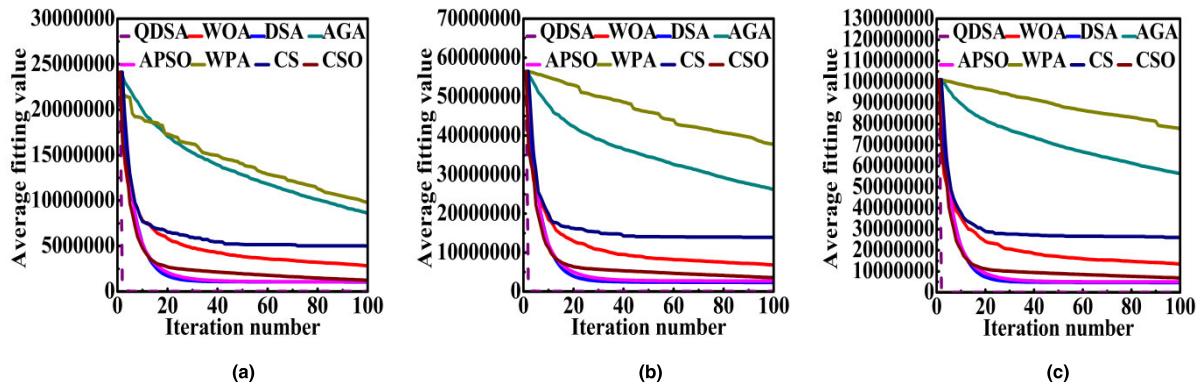


FIGURE 14. Convergence curve based on QDSA and advanced algorithms considered for comparison using the different dimension of the F6: (a) 200 dimension; (b) 300 dimension; (c) 400 dimension.

V. DISCUSSION

A. RESEARCH FINDINGS

The solution of large-scale functions has always been a hot topic for scholars at home and abroad. To solve the problem of large-scale function more accurately, a new algorithm (QDSA), is proposed in this study. Based on six commonly used large-scale test functions, the performance of QDSA and advanced algorithms are compared. Some interesting findings are as follows:

(1) According to Min, Max, Mean, Std., A.G, QDSA is better than DSA based on the different dimension of the Rastrigin function, Ackley function, Griewank function, Sphere function, LEVY function and ROTATED HYPER-ELLIPSOID function. This finding shows that the combination of quantum search and dolphin swarm algorithm effectively improves the problem of obtaining a globally optimal solution by DSA.

(2) Compared with WOA, AGA, APSO, WPA, CS, and CSO, QDSA has a stronger ability to obtain the optimal solution. This finding implies that QDSA is an effective improvement.

B. FUTURE RESEARCH

Although the proposed QDSA is better, the parameters are fixed when the QDSA is tested, which limits the ability of QDSA to solve large-scale functions to a certain extent. So, in future research, an adaptive parameter change of QDSA is a research direction. At the same time, the innovative method is applied to other fields to further verify the practical application value of the method, e.g. energy storage optimization [39]–[44], the parameter optimization of natural gas load prediction [45]–[50] etc.

VI. CONCLUSION

Aiming at the problem that DSA has weak global convergence ability and is easy to fall into local optimum, this paper proposed QDSA. It is compared with CS, DSA, AGA, APSO, WPA, WOA, and CSO. Through experiments, the performance of QDSA in solving large-scale functions is better than that of other algorithms considered for comparison.

Besides, QDSA improves the problem of DSA falling into the local solution and QDSA obtains optimal solutions for different scale test functions. Therefore, it can be concluded that QDSA is an effective improvement.

REFERENCES

- [1] A. Kaveh, *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Cham, Switzerland: Springer, 2017.
- [2] A. Kaveh and M. I. Ghazaan, *Meta-Heuristic Algorithms for Optimal Design of Real-Size Structures*. Cham, Switzerland: Springer, 2018.
- [3] S. M. Erfani, S. Rajasegarar, C. Leckie, and S. Karunasekera, “High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning,” *Pattern Recognit.*, vol. 58, pp. 121–134, Oct. 2016.
- [4] C. Sun, Y. Jin, J. Ding, J. Zeng, and R. Cheng, “Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems,” *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 644–660, Aug. 2017.
- [5] S. Gu, R. Cheng, and Y. Jin, “Feature selection for high-dimensional classification using a competitive swarm optimizer,” *Soft Comput.*, vol. 22, no. 3, pp. 811–822, Feb. 2018.
- [6] W. Yu, X. Qu, and Y. Bo, “Improved chicken swarm optimization method for reentry trajectory optimization,” *Math. Problems Eng.*, vol. 2018, Jan. 2018, Art. no. 8135274.
- [7] Y. Khoo, J. Lu, and L. Ying, “Solving for high-dimensional committer functions using artificial neural networks,” *Res. Math. Sci.*, vol. 6, p. 1, Mar. 2019.
- [8] H. Lu, K. Huang, L. Guo, and M. Azimi, “Blockchain technology in the oil and gas industry: A review of applications, opportunities, challenges, and risks,” *IEEE Access*, vol. 7, pp. 41426–41444, 2019, doi: [10.1109/ACCESS.2019.2907695](https://doi.org/10.1109/ACCESS.2019.2907695).
- [9] D. Haifeng, G. Moaguo, L. Ruochen, and J. Licheng, “A novel algorithm of artificial immune system for high-dimensional function numerical optimization,” *Prog. Natural Sci.*, vol. 15, no. 5, pp. 463–471, 2005.
- [10] Z. Yang, K. Tang, and X. Yao, “Differential evolution for high-dimensional function optimization,” in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 3523–3530.
- [11] Y. Chu, H. Mi, Z. Ji, Q. H. Wu, and H. Liao, “A fast bacterial swarming algorithm for high-dimensional function optimization,” in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2008, pp. 3135–3140.
- [12] L. Xiaohu, Z. Jinhua, W. Sunan, L. Maolin, and L. Kunpeng, “A small world algorithm for high-dimensional function optimization,” in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Automat.*, Dec. 2009, pp. 55–59.
- [13] Y. Ren and Y. Wu, “An efficient algorithm for high-dimensional function optimization,” *Soft Comput.*, vol. 17, no. 6, pp. 995–1004, 2013.
- [14] C. Wang and J.-H. Gao, “A differential evolution algorithm with cooperative coevolutionary selection operation for high-dimensional optimization,” *Optim. Lett.*, vol. 8, no. 2, pp. 477–492, 2014.
- [15] M. Moshki, P. Kabiri, and A. Mohebalhojeh, “Scalable feature selection in high-dimensional data based on GRASP,” *Appl. Artif. Intell.*, vol. 29, no. 3, pp. 283–296, 2015.

- [16] A. A. A. Esmin, R. A. Coelho, and S. Matwin, "A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data," *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 23–45, 2015.
- [17] J. Han, E. JentzenWeinan, and A. Jentzen, "Solving high-dimensional partial differential equations using deep learning," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 34, pp. 8505–8510, 2018.
- [18] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Inf. Sci.*, vols. 454–455, pp. 59–72, Jul. 2018.
- [19] L. Wen, X. Liang, M. Tang, and J. Jiao, "Inspired grey wolf optimizer for solving large-scale function optimization problems," *Appl. Math. Model.*, vol. 60, pp. 112–126, Aug. 2018.
- [20] S. Özgün, C. Yaşar, and H. Temurtaş, "Incremental gravitational search algorithm for high-dimensional benchmark functions," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3779–3803, 2018.
- [21] P. S. Shelokar, P. Siarry, B. D. Kulkarni, and V. K. Jayaraman, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," *Appl. Math. Comput.*, vol. 188, no. 1, pp. 129–142, 2007.
- [22] Y.-J. Wang, "Improving particle swarm optimization performance with local search for high-dimensional function optimization," *Optim. Methods Softw.*, vol. 25, no. 5, pp. 781–795, 2010.
- [23] S. Mirjalili, G.-G. Wang, and L. Dos S. Coelho, "Binary optimization using hybrid particle swarm optimization and gravitational search algorithm," *Neural Comput. Appl.*, vol. 25, no. 6, pp. 1423–1435, 2014.
- [24] Z. Li, W. Wang, Z. Li, and Y. Yan, "PS-ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems," *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8881–8895, 2015.
- [25] J. Luo and B. Shi, "A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems," *Appl. Intell.*, vol. 49, no. 5, pp. 1982–2000, 2019.
- [26] Z.-J. Teng, J.-L. Lv, and L.-W. Guo, "An improved hybrid grey wolf optimization algorithm," *Soft Comput.*, vol. 23, no. 15, pp. 6617–6631, 2019.
- [27] I. B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Appl. Soft Comput.*, vol. 66, pp. 232–249, May 2018.
- [28] Z. Liang, F. Yang, and J. Ouyang, "A hybrid GA-PSO optimization algorithm for conformal antenna array pattern synthesis," *J. Electromagn. Waves Appl.*, vol. 32, no. 13, pp. 1601–1615, 2018.
- [29] G. I. Sayed and A. E. Hassanien, "A hybrid SA-MFO algorithm for function optimization and engineering design problems," *Complex Intell. Syst.*, vol. 4, no. 3, pp. 195–212, 2018.
- [30] R. Dong, S. Wang, G. Wang, and X. Wang, "Hybrid optimization algorithm based on wolf pack search and local search for solving traveling salesman problem," *J. Shanghai Jiaotong Univ. (Sci.)*, vol. 24, no. 1, pp. 41–47, 2019.
- [31] W. Qiao and Z. Yang, "Modified dolphin swarm algorithm based on chaotic maps for solving high-dimensional function optimization problems," *IEEE Access*, vol. 7, pp. 110472–110486, 2019. doi: [10.1109/ACCESS.2019.2931910](https://doi.org/10.1109/ACCESS.2019.2931910).
- [32] A. Kaveh, S. R. H. Vaez, and P. Hosseini, "Modified dolphin monitoring operator for weight optimization of frame structures," *Periodica Polytechnica Civil Eng.*, vol. 61, no. 4, pp. 770–779, 2017.
- [33] H. Arzani, A. Kaveh, and M. Kamalinejad, "Optimal design of pitched roof rigid frames with non-prismatic members using quantum evolutionary algorithm," *Periodica Polytechnica Civil Eng.*, vol. 63, no. 2, pp. 593–607, 2019.
- [34] M. Kamalinejad, H. Arzani, and A. Kaveh, "Quantum evolutionary algorithm with rotational gate and H_ϵ -gate updating in real and integer domains for optimization," *Acta Mechanica*, vol. 230, no. 8, pp. 2937–2961, 2019.
- [35] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
- [36] J. Santos, A. Ferreira, and G. Flintsch, "An adaptive hybrid genetic algorithm for pavement management," *Int. J. Pavement Eng.*, vol. 20, no. 3, pp. 266–286, 2019.
- [37] D. Chitara, K. R. Niazi, N. Gupta, and A. Swarmkar, "Cuckoo search optimization algorithm for designing of a multimachine power system stabilizer," *IEEE Trans. Ind. Appl.*, vol. 54, no. 4, pp. 3056–3065, Jul./Aug. 2018.
- [38] J. Liang, L. Wang, M. Ma, and J. Zhang, "A fast SAR image segmentation method based on improved chicken swarm optimization algorithm," *Multimedia Tools Appl.*, vol. 77, no. 24, pp. 31787–31805, 2018.
- [39] S. S. Reddy, A. R. Abhyankar, and P. R. Bijwe, "Market clearing for a wind-thermal power system incorporating wind generation and load forecast uncertainties," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2012, pp. 1–8.
- [40] S. S. Reddy and C. S. Rathnam, "Optimal power flow using glow-worm swarm optimization," *Int. J. Elect. Power Energy Syst.*, vol. 80, pp. 128–139, Sep. 2016.
- [41] S. S. Reddy and P. R. Bijwe, "Real time economic dispatch considering renewable energy resources," *Renew. Energy*, vol. 83, pp. 1215–1226, Nov. 2015.
- [42] S. S. Reddy, "Optimal scheduling of thermal-wind-solar power system with storage," *Renew. Energy*, vol. 101, pp. 1357–1368, Feb. 2017.
- [43] S. S. Reddy, V. Sandeep, and C.-M. Jung, "Review of stochastic optimization methods for smart grid," *Frontiers Energy*, vol. 11, no. 2, pp. 197–209, 2017.
- [44] S. S. Reddy, "Optimization of renewable energy resources in hybrid energy systems," *J. Green Eng.*, vol. 7, no. 1, pp. 43–60, 2017.
- [45] W. Qiao, K. Huang, S. Han, and M. Azimi, "A novel hybrid prediction model for hourly gas consumption in supply side based on improved whale optimization algorithm and relevance vector machine," *IEEE Access*, vol. 7, pp. 88218–88230, 2019.
- [46] Q. Weibiao, L. Bingfan, and K. Zhangyang, "Differential scanning calorimetry and electrochemical tests for the analysis of delamination of 3PE coatings," *Int. J. Electrochem. Sci.*, vol. 14, pp. 7389–7400, Aug. 2019.
- [47] W. Qiao and H. Wang, "Analysis of the wellhead growth in HPHT gas wells considering the multiple annuli pressure during production," *J. Natural Gas Sci. Eng.*, vol. 50, pp. 43–54, Feb. 2018.
- [48] W. Liu, Z. Zhang, J. Chen, J. Fan, D. Jiang, Y. Li, and D. Jjk, "Physical simulation of construction and control of two butted-well horizontal cavern energy storage using large molded rock salt Specimens," *Energy*, vol. 185, pp. 682–694, Oct. 2019.
- [49] E. Liu, W. Li, H. Cai, and S. Peng, "Formation mechanism of trailing oil in product oil pipeline," *Processes*, vol. 7, no. 1, p. 7, 2019.
- [50] Z. Su, E. Liu, Y. Xu, P. Xie, C. Shang, and Q. Zhu, "Flow field and noise characteristics of manifold in natural gas transportation station," *Oil Gas. Sci. Technol.*, vol. 74, no. 70, Jul. 2019. doi: [10.2516/ogst/2019038](https://doi.org/10.2516/ogst/2019038).



WEIBIAO QIAO received the B.S. degree in information and computing science from Northeast Agricultural University, Harbin, Heilongjiang, China, in 2009, the M.S. degree in oil and gas storage and transportation engineering from Liaoning Shihua University, Fushun, Liaoning, China, in 2012, and the Ph.D. degree in oil and gas storage and transportation engineering from China Petroleum University, Qingdao, Shandong, China, in 2017.

Since 2017, he has been holding a Lectureship at the North China University of Water Resources and Electric Power, Zhengzhou, Henan, China, and is involved in Postdoctoral research with Sinopec Petroleum Engineering Zhongyuan Design Co., Ltd. His research interests include gas consumption forecasting, gas pipeline robot detection, and intelligent scheduling of gas storage group, and so on.



ZHE YANG was born in Shanxi, China. He received the B.S. degree in computer science from the Taiyuan University of Technology (TUT), Shanxi, in 2017. He is currently pursuing the M.Eng. degree with the School of Computer Science, The University of Manchester, Manchester, U.K. He is involved in swarm intelligence algorithm and machine learning.